

Spojování tabulek

Spojování tabulek

- Pomocí spojování může příkaz **SELECT** vybírat data z více tabulek.
- Základní způsob spojení je vyjmenovat v příkazu **SELECT** v klauzuli **FROM** více tabulek:
 - **SELECT** * **FROM** zamestnanec, zamestnani;
- Pokud není uvedena klauzule **WHERE** je výsledkem kartézský součin tabulek – sada, v níž budou všechny kombinace řádků z obou (případně ze všech) tabulek. Pomocí klauzule **WHERE** se omezí výsledná sestava jen na řádky, které spolu logicky souvisí (vnitřní spojení tabulek):
 - **SELECT** * **FROM** zamestnanec, zamestnani
WHERE zamestnanec.zamestnani_id =
zamestnani.zamestnani_id;

Spojování tabulek

- Použití * v dotazu vrátí všechny sloupce ze všech tabulek.
- Je možné vrátit data z jedné tabulky nebo jen vybrané atributy:
 - **SELECT** <tabulka>.* **FROM** <seznam tabulek>;
 - **SELECT** <tabulka>.<atribut>
FROM <seznam tabulek>;
 - **SELECT** zamestnanec.*, zamestnani.nazev_pozice
FROM zamestnanec, zamestnani;
- Pokud dvě nebo více tabulek obsahují sloupce se stejnými názvy a do výsledné sestavy jsou potřeba zahrnout, lze je rozlišit pomocí aliasů. Aliasy lze připojovat i k tabulkám:
 - **SELECT** t1.nazev AS nazev_typu,
 d.nazev nazev_destinace
FROM typ_letadla t1, destinace d;

Spojování tabulek

- Když se ponechají stejné názvy dvou a více sloupců ve výsledné sestavě, záleží na zvolené databázi, jak budou ve výsledku tyto sloupce pojmenovány:
 - MySQL ponechává názvy sloupců stejné,
 - Oracle sloupce se shodným názvem očísluje – nazev, nazev_1, nazev_2

Hvězdička v příkazu SELECT

- Použití * v příkazu SELECT se nedoporučuje. Lepší variantou je vyjmenování sloupců, které budou ve výsledné sestavě. Důvody:
 - Ke hvězdičce nelze přiřadit alias,
 - rychlost dotazu – pro tabulku s velkým množstvím sloupců, může být výsledná sestava objemná,
 - lepší ladění a odolnost aplikace – pokud příkaz spoléhá na existenci sloupce <název sloupce>, ale tento sloupec je odstraněn nebo přejmenován (změna struktury tabulky) příkaz s hvězdičkou přesto projde, ale chyba nejspíš nastane někde později v aplikaci. Pokud by SELECT požadoval přímo daný sloupec, skončí již dotaz do databáze chybou.

Spojování tabulek

Spojení tabulek lze rozdělit na:

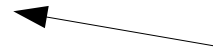
- Vnitřní – jsou zobrazena pouze data, která si v tabulkách odpovídají. Mezi vnitřní spojení patří:
 - CROSS JOIN,
 - INNER JOIN.
- Vnější – vrací všechny záznamy, které vyhovují podmínce spojení. Navíc jsou zobrazena i data, která mají záznamy v jedné tabulce, ale v druhé tabulce ne. Takové sloupce, které jsou získávány z druhé tabulky, budou NULL. Mezi vnější spojení patří:
 - LEFT JOIN,
 - RIGHT JOIN,
 - FULL JOIN.
- Přirozené spojení NATURAL JOIN – může být vnitřní i vnější.

Vnitřní spojení - CROSS JOIN

- CROSS JOIN – tabulky se spojí tak, že výsledkem je jejich kartézský součin. Obdoba základního vnitřního spojení.
- Pro každý řádek z jedné tabulky je ve výsledné sadě záznamů vytvořena kombinace se všemi řádky z tabulky druhé.
- Příklad:

```
SELECT d.destinace_id, z.poradi_zastavky FROM  
destinace d CROSS JOIN zastavka z;  
-- 2.varianta  
SELECT d.destinace_id, z.poradi_zastavky FROM  
destinace d, zastavka z;
```

Vnitřní spojení – INNER JOIN

- INNER JOIN – některé řádky, které nemají odpovídající záznamy v jiných tabulkách, se do výsledné souhrnné tabulky (sestavy) vůbec nedostanou. Příklad:
 - `SELECT jmeno, prijmeni, nazev_pozice
FROM zamestnanec zc, zamestnani zi
WHERE zc.zamestnani_id = zi.zamestnani_id;
-- 2.varianta`
 - `SELECT jmeno, prijmeni, nazev_pozice
FROM zamestnanec zc INNER JOIN zamestnani zi
ON zc.zamestnani_id = zi.zamestnani_id;`
- Klauzuli USING lze použít, pokud se sloupce (podle kterých se spojuje) jmenují shodně:
 - `SELECT jmeno, prijmeni, nazev_pozice
FROM zamestnanec zc INNER JOIN zamestnani zi
USING(zamestnani_id);` 
- Poznámka: Klíčové slovo INNER je nepovinné.

Vnější spojení - klauzule JOIN

- Syntaxe dotazu vnějšího spojení:

```
SELECT <seznam výstupních sloupců>
FROM [<seznam tabulek>,] <spojovaná tabulka>
{
    {FULL|LEFT|RIGHT} [OUTER] JOIN <spoj.tabulka>
    { ON <podmínka spojení> | USING (<sloupce>) }
    |
    NATURAL {FULL|LEFT|RIGHT} [OUTER]
    JOIN <spojovaná tabulka>
} [, <seznam tabulek>]
[WHERE <podmínka pro výběr řádků>]
[GROUP BY <seznam sloupců podle kt. se seskupuje>]
[HAVING <podmínka pro seskupení>]
[ {UNION|UNION ALL|INTERSECT|MINUS} <příkaz_select> ]
[ORDER BY <seznam řazených sloupců>];
```

- Poznámka: Klíčové slovo **OUTER** je nepovinné.

Vnější spojení – LEFT JOIN

- Pro spojení tabulek **A** a **B**, kde je třeba mít zahrnuté všechny záznamy z **A**, i když tabulka **B** neobsahuje odpovídající záznamy, je vhodné použít vnější spojení LEFT JOIN.
- Příklad: Spojení tabulek **A** a **B**:

```
SELECT A.nazev AS nazev_A, B.nazev AS nazev_B  
FROM A LEFT JOIN B ON A.b_id = B.id;
```

Tabulka A
NAZEV | B_ID

A1	4
A2	null
A3	1
A4	5

Tabulka B
ID | NAZEV

1	B1
2	B2
3	B3
4	B4

Výsledek
NAZEV_A | NAZEV_B

A1	B4
A2	null
A3	B1
A4	null

Příklady spojení různých matic pomocí LEFT JOIN:

LEFT JOIN:

A:

A	1
B	2
C	3
D	4

B:

1	B1
2	B2

A	1	1	B1
B	2	2	B2
C	3	null	null
D	4	null	null

A LEFT JOIN B

A	1
B	2
C	3
D	4

1	B1
1	B2

A	1	1	B1
A	1	1	B2
B	2	null	null
C	3	null	null
D	4	null	null

LEFT JOIN:

A

A1	1
A2	2
A3	3
A4	4
A5	5
A6	null

B

1	B1
3	B2
6	B3
10	B4
null	B5

A1	1	1	B1
A2	2	null	null
A3	3	3	B2
A4	4	null	null
A5	5	null	null
A6	null	null	null

Vnější spojení – RIGHT JOIN

- Pro spojení tabulek **A** a **B**, kde je třeba mít zahrnuté všechny záznamy z **B**, i když tabulka **A** neobsahuje odpovídající záznamy, je vhodné použít vnější spojení RIGHT JOIN.
- Př.: Spojení tabulek **A** a **B**:

```
SELECT A.nazev AS nazev_A, B.nazev AS nazev_B  
FROM A RIGHT JOIN B ON A.b_id = B.id;
```

Tabulka A
NAZEV | B_ID

A1	4
A2	null
A3	1
A4	5

Tabulka B
ID | NAZEV

1	B1
2	B2
3	B3
4	B4

Výsledek
NAZEV_A | NAZEV_B

A3	B1
null	B2
null	B3
A1	B4

Opět jako u vnitřního spojení platí, že poslední tabulka (D) se připojuje k aktuálně vytvořené matici (A + B + C):

select * from A join B join C left join D



(A) LEFT JOIN B

~
B RIGHT JOIN (A)

Full join: A LEFT JOIN B RIGHT JOIN B

Úplné vnější spojení – FULL JOIN

- Kombinace RIGHT a LEFT JOIN spojení.
- Výsledná sestava obsahuje všechny záznamy z tabulky A i B. Tam, kde chybí záznamy v jedné nebo druhé tabulce jsou uvedeny null hodnoty.
- Př.: Spojení tabulek **A** a **B**:

```
SELECT A.nazev AS nazev_A, B.nazev AS nazev_B  
FROM A FULL JOIN B ON A.b_id = B.id;
```

Tabulka A	
NAZEV	B_ID

A1	4
A2	null
A3	1
A4	5

Tabulka B	
ID	NAZEV

1	B1
2	B2
3	B3
4	B4

Výsledek	
NAZEV_A	NAZEV_B

A1	B4
A2	null
A3	B1
A4	null
null	B2
null	B3

Přirozené spojení – NATURAL JOIN

- NATURAL JOIN – tabulky se spojí podle sloupců se stejným názvem. Př.:
 - **SELECT** jmeno, prijmeni, nazev_pozice
FROM zamestnanec **NATURAL JOIN** zamestnani;
 - Předpokládá se, že obě tabulky obsahují sloupec zamestnani_id a podle něj se spojuje.
- Nevýhody použití NATURAL JOINu:
 - Při přidání sloupce se stejným názvem do jedné tabulky, který je již obsažen v druhé spojované tabulce, přestane dotaz vracet správná data – složité hledání chyby.
 - Na první pohled není znatelné, podle kterých sloupců se tabulky spojují.

Oracle operator (+)

- Operátor (+) má stejnou funkci jako LEFT nebo RIGHT JOIN, ale jedná se o starší způsob zápisu vnějšího spojení. Příklad použití:
- Původní dotaz:

```
SELECT jmeno, prijmeni, nazev_pozice  
FROM zamestnani LEFT JOIN zamestnanec  
ON zamestnani.zamestnani_id = zamestnanec.zamestnani_id;
```
- S použitím (+):

```
SELECT jmeno, prijmeni, nazev_pozice  
FROM zamestnani, zamestnanec  
WHERE zamestnani.zamestnani_id = zamestnanec.zamestnani_id (+);
```
- Operátor (+) se uvede u tabulky, ve které nemusí být odpovídající záznamy.
- Poznámka: operátor (+) není možné kombinovat s novým způsobem spojování tabulek.

Nový nebo starý způsob spojení?

- **Optimalizační tip:**
 - Ze syntaktického pohledu při starém způsobu spojení (Oracle syntax → operátor (+)) může dojít k záměně spojení (join) a omezení (restriction).

SELF JOIN

- Můžete spojit tabulku samu na sebe. Abyste mezi jednotlivými tabulkami rozlišili, je nutné nastavit jejich alias.
- Výpis všech jmen zaměstnanců a jmen jejich nadřízených:

```
SELECT
    a.jmeno || ' ' || a.prijmeni zamestnanec,
    b.jmeno || ' ' || b.prijmeni nadrizeny
FROM zamestnanec a
LEFT JOIN zamestnanec b
ON a.nadrizeny = b.zamestnanec_id
```

- U Self joinu se musí vždy použít JOIN ... ON syntaxe, není možné použít JOIN ... USING ani NATURAL JOIN.